

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Simon Vesel

**Mobilni telefon kot pripomoček pri
športni vadbi**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Viljan Mahnič

Ljubljana 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Proučite možnosti, kako bi lahko pametni telefon uporabili kot pripomoček pri športni vadbi, npr. pri izvajanju razteznih vaj. Zahteve uporabnikov definirajte v obliki uporabniških zgodb in opišite postopek njihove realizacije s poudarkom na uporabljenih tehnologijah. Izdelana aplikacija naj bo na voljo tako za pametne telefone kot za tablične računalnike, njen izgled pa naj bo optimalen ne glede na velikost naprave in njen položaj (pokončno/ležeče).

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Simon Vesel, z vpisno številko **63090161**, sem avtor diplomskega dela z naslovom:

Mobilni telefon kot pripomoček pri športni vadbi

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Viljana Mahniča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15. septembra 2014

Podpis avtorja:

Zahvaljujem se mentorju izr. prof. dr. Viljanu Mahniču za vodenje in pomoč pri izdelavi diplomskega dela.

Zahvaljujem se tudi družini in Špeli za vso podporo in razumevanje v času mojega študija.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Zahteve za aplikacijo	5
2.1	Raztezanje	5
2.2	Nabor razteznih vaj z opisi	8
2.3	Kaj pričakujemo od aplikacije	24
3	Opis uporabljenih orodij in tehnologij	27
3.1	Android	27
3.2	Android SDK	29
3.3	Android developer tools	29
3.4	Programski jezik Java	30
3.5	Podatkovni sistem SQLite	31
3.6	GenyMotion	31
4	Programska rešitev mobilne aplikacije	33
4.1	Poimenovanje in logotip	33
4.2	Nastavitve ADT	33
4.3	Podatkovna baza	35
4.4	Realizacija uporabniške zgodbe 1	35
4.5	Realizacija uporabniške zgodbe 2	39

KAZALO

4.6	Realizacija uporabniške zgodbe 3	40
4.7	Realizacija uporabniške zgodbe 4	41
4.8	Realizacija uporabniške zgodbe 5	42
4.9	Realizacija uporabniške zgodbe 6	45
4.10	Opis uporabe celotne aplikacije	45
5	Sklepne ugotovitve	49
	Literatura	51

Povzetek

Cilj diplomske naloge je bil pregled razvoja mobilnih telefonov, današnjega stanja in zmožnosti uporabe v športne namene. Osredotočili smo se na raztežno vadbo. Pregledali smo, kako redno izvajanje raztezanja učinkuje na človeka ter zapisali splošna vodila za vadbo. V ta namen smo pozneje razvili mobilno aplikacijo Tendo. Za razvoj smo si izbrali mobilni operacijski sistem Android. Najprej smo spoznali delovanje le tega in se seznanili s potrebno programsko opremo. Sledil je razvoj aplikacije, ki športnikom olajša izvajanje raztezanja. Omogoča jim poljuben izbor razteznih vaj, opremljenih s slikovnim materialom in tekstovnim opisom, nato pa jih s pomočjo zvočnih opozoril vodi skozi proces vadbe. Predstavili smo tudi nekaj ključnih delov programske kode, potrebnih za samo realizacijo.

Ključne besede: Android, raztezanje, mobilni telefon, mobilna aplikacija, šport.

Abstract

The aim of this bachelor thesis was the overview of mobile phone evolution, current state and its usage in sport. We focused on stretching, presenting a group of exercises with photos and descriptions. We wrote about the effects regular stretching workout has on a person and pointed out a few guidelines on how to do stretching exercises properly. We also took a look into Android and other programs we used to develop our application. We then developed a mobile application named Tendo that facilitates this type of workout. Application guides the user through the workout using sound notifications. We included and explained the most important parts of source code.

Keywords: Android, stretching, mobile phone, mobile application, sport.

Poglavje 1

Uvod

Vse od prve pojavitve takoimenovanega prenosnega telefona pa do danes je minilo že mnogo let [5, 7]. Čeprav so se mobilni telefoni pojavili že prej in sicer tako, da so bili vgrajeni v avtomobile, pa so se v ročni obliki, kot jo poznamo danes, pojavili šele leta 1973. Tretjega aprila tega leta je raziskovalec Martin Cooper opravil prvi klic z napravo s takrat, za današnje razmere nepredstavljivo okornimi merami in težo dobrega kilograma, baterijsko napajanje pa je omogočalo vsega 20 minut delovanja. Prvi mobilni telefon, Motorola DynaTAC 8000x [4], dostopen za javno uporabo se je pojavil desetletje pozneje. Navkljub več tisoč dolarjev visoki ceni, še vedno zelo kratki dobi delovanja na baterijo in okorni izdelavi, je bilo zanimanje zanj ogromno. Skozi leta pa se je z nadaljnim razvojem in nižanjem cen število uporabnikov le še večalo in dandanes je na svetu že skoraj toliko mobilnih telefonov, kot je celotne populacije [9].

Nekdaj veliki, energijsko potratni in predvsem nerodni telefoni so do danes napredovali v majhne, kompaktne, večnamenske in oblikovno dovršene naprave. Če je bila na začetku osnovna in edina funkcija telefona klic ali pogovor med dvema osebama, je danes vse prej kot to. Najprej se je telefonskim klicom pridružilo pošiljanje kratkih sporočil, razvoj interneta in podatkovni prenos pa je uporabnikom mobilnega telefona odprl povsem nova vrata. Dostopanje do najrazličnejših informacij je tako postalo enostavno na vsakem



Slika 1.1: Motorola DynaTAC 8000x

koraku. Hitro lahko poizvemo kateri film se danes predvaja v bližnji kinoteki, pogledamo kakšno vreme nas čaka jutri ali pa si ogledamo foto galerijo s koncerta, ki se je odvijal prejšnji vikend. Z dodano kamero, infrardečim in bluetooth vmesnikom lahko zabeležimo dragocene trenutke, upravljamo televizijski sprejemnik ali se povežemo z napravami v bližini. Povrh vsega pa novejši mobilni telefone krasi še kopica vgrajenih senzorjev, ki napravo naredijo celovito in pripravljeno za res raznoliko uporabo.

Eden izmed teh načinov uporabe, pravzaprav vse bolj popularen, je zagotovo v športne namene [10]. Vse to se je začelo z osnovnim merjenjem časa. Znameniti model 3310 [6] podjetja Nokia je bil eden prvih, ki je med svojimi funkcijami nudil tudi štoparico. Skozi leta je tehnološki napredek mobilne telefone opremil s kopico senzorjev *cites5sensors*. S pospeškometerom lahko zaznamo smer gibanja telefona v treh oseh, podobno vlogo ima tudi žiroskop, ki pa je bolj natančen in zmožen zaznati tudi rotacijo telefona. Z barometrom lahko zaznamo na kakšni nadmorski višini se nahajamo. Med tem, ko pedometer šteje naše korake, lahko merimo naš utrip z merilnikom srčnega utripa. Traso našega pohoda ali teka pa lahko s pomočjo globalnega sistema

pozicioniranja (GPS) in namenske programske opreme zabeležimo. Mobilni telefoni pa z glasbenimi predvajalniki poskrbijo, da vadba mine hitreje.

Če se je pred dobrimi štiridesetimi leti vse skupaj pričelo z enostavnim telefonskim pogovorom, pa ima danes uporabnik v majhni napravi že kar svojega osebnega športnega trenerja, ki bdi nad njegovo vadbo, spremlja njegovo pripravljenost in mu pomaga pri doseganju boljših športnih rezultatov.

Razvoj vseh teh senzorjev in dobre programske opreme je uporabniku, poleg vseh standardnih funkcij kot so klicanje, pošiljanje kratkih sporočil in brskanje po spletu, omogočil natančen pregled nad lastno vadbo in pomoč pri izvajanju le te. S podobnim namenom smo se tudi lotili pisanja te diplomske naloge. Za cilj smo si zadali izdelavo preproste aplikacije za mobilne telefone, v našem primeru za mobilno platformo Android, ki športnikom pomaga pri izvajanju raztezanja. Raztezanje je namreč nepogrešljiv del treninga vsakega športnika, bodisi profesionalnega bodisi rekreativnega. Poleg tega pa raztezanje pozitivno učinkuje na vsakogar, pa naj bo to pretegovanje na delovnem mestu ali pa ob jutranjem bujenju.

V naslednjem poglavju bomo tako opredelili zahteve za izdelavo mobilne aplikacije. Naredili bomo pregled nad vključenimi razteznimi vajami in opisali pričakovane funkcionalnosti aplikacije. Sledil bo opis tehnoloških rešitev, ki bodo pripomogle pri končni izvedbi. V četrtem poglavju bo z razlago prikazana realizacija same aplikacije, na koncu pa bomo obravnavali še rezultate izvedbe.

Poglavje 2

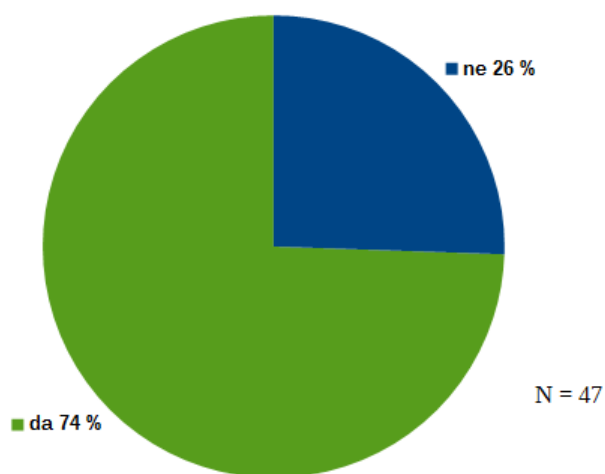
Zahteve za aplikacijo

Preden se lotimo same izvedbe je dobro, da poznamo teoretično vsebino aplikacije ter tehnološke rešitve, s katerimi bomo prišli do končnega rezultata. Da pa izvedba ne bo zaman, se splača preveriti zanimanje s kratko anketo. Tako smo smiselnost izdelave preverili s kratko anketo med športniki. 50 oseb smo povprašali ali izvajajo raztezanje ter ali se jim predlagana rešitev, izvajanje raztezanja s pomočjo mobilne aplikacije, zdi smiselna in bi bila za njih uporabna. Rezultati so bili spodbudni. Izmed 50 oseb jih 47 izvaja raztezanje, izmed teh pa je na vprašanje o interesu za aplikacijo 35 oseb odgovorilo pritrdilno. Tako smo zaznali preko 70 odstoten interes, kar je bila lepa popotnica za nadaljnje delo.

Poleg zanimanja za aplikacijo smo preverili še kateri mobilni operacijski sistem uporabljajo anketiranci. Rezultati so pokazali, da nekaj manj kot 80 odstotkov anketirancev uporablja Android. To je bila, poleg splošne statistike o uporabi operacijskega sistema Android [11], še ena potrditev, da s pravim izdelkom ciljamo pravi operacijski sistem.

2.1 Raztezanje

Čeprav so koristi raztezanja izjemne [3], se še vedno najde kar veliko športnikov in rekreativcev, ki temu delu treninga posvetijo premalo pozornosti. »Od



Slika 2.1: Rezultati ankete - interes za aplikacijo

tega se ne dobi mišic« je samo eden izmed raznih izgovorov. Čeprav gre morda za neprivlačen del treninga, ta vendarle prinaša precej koristi in je eden izmed nujnih za vse tiste, ki stremijo k napredku in so temu primerno pripravljeni nameniti čas.

2.1.1 Zakaj raztezati

Dober voznik s svojim avtomobilom nikoli ne dirja takoj, ko ga prižge, temveč počaka, da se motor ogreje in pripravi na delo. Raztezanje ima v športu podobno vlogo. Z raztezanjem mišice pripravimo na delo, poleg tega pa vzdržujemo dobro gibljivost. Bolj kot je mišica prožna, bolje se počuti in ima večjo možnost za rast. Poleg tega pa z raztezanjem izboljšamo krvni obtok, zato se mišice napolnijo s svežjo krvjo, bogato s kisikom. To osveži in poživiti telo. Po raztezanju smo tako lahki, močni, gibčni in živahni.

2.1.2 Prednosti rednega raztezanja

Redno izvajanje raztezanja ima številne pozitivne učinke [2, 18]:

- izboljšuje telesno držo, koordinacijo in gibljivost,
- skrbi za hitro obnovo organizma, zlasti po intenzivni športni vadbi,
- spodbuja krvni obtok, izboljša prekrvavitev v mišicah, ki postanejo zmogljivejše,
- zmanjšuje napetost v mišicah, s čimer pripomore k sprostitvi celotnega telesa,
- izboljšuje splošno gibljivost in telesno zmogljivost,
- razvija boljše občutenje telesa,
- izboljša mentalno budnost.

Z raztezanjem bodo sklepi vse bolj gibljivi in vezi bolj prožne, mišice pa bodo prijetno napete. Poleg tega pa bomo še sprostili morebitno neustrezno skrčenost in napetost v mišicah.

2.1.3 Kdaj in kako raztezati

Za optimalno izvajanje raztezanja se je priporočljivo držati naslednjih vodil [1, 18]:

- Zjutraj, ko vstanemo, naredimo nekaj razteznih vaj. S tem bomo spodbudili prekrvavitev in se hitreje popolnoma prebudili.
- Raztezajmo se tudi na delovnem mestu ter po dolgem stanju ali sedenju. Tako bomo učinkovito sprostili stres in napetost.
- Kadarkoli začutimo otrdelost, razbolelost ali utrujenost.
- Kadarkoli se želimo osredotočiti in delati po najboljših močeh
- Preden se lotimo aerobne vadbe (tek, kolesarjenje, ...) naredimo nekaj razteznih vaj, vsaj za vse večje mišice nog, v raztegih pa vztrajajmo le kratek čas.

- Raztegnimo vse mišice, ki bodo med vadbo obremenjene - s tem jih bomo pripravili na večji napor.
- Za sprostitev kronično napetih in skrajšanih mišic je priporočljivo, da raztezanje izvajamo pogosteje.
- Pred intenzivnejšim raztezanjem se je potrebno rahlo ogreti.

2.1.4 Zapovedi in prepovedi raztezanja

Da bo izvajanje raztezanja kar se da pravilno, se moramo držati nekaterih pravil [1, 18]. Ob raztezanju se moramo vedno sprostiti in dihati lahko. Prisluhniti moramo telesu in se osredotočiti na mišice in sklepe, katere raztezamo. Občutiti moramo raztezanje in obseg le tega prilagajati po občutku. Nikoli ne smemo izvajati sunkovitih gibov. Za naglico med raztezanjem ni prostora. Med vadbo ne smemo biti napeti, prav tako ne smemo zadrževati sape. Mišic ne smemo raztezati do bolečine. V kolikor začutimo bolečino, moramo popustiti. Dovolj je, da začutimo rahlo napetost mišice. Ta naj bo taka, da se počutimo lagodno. V kolikor smo poškodovani, se moramo odpraviti k zdravniku, saj te vaje niso namenjene za zdravljenje hujših poškodb in težav. Pri izvajanju raztezanja moramo zato biti previdni, saj utegnemo storiti več škode kot koristi.

2.2 Nabor razteznih vaj z opisi

Pred nami je nabor dvajsetih vaj za raztezanje. Vaje bodo predstavljene v kombinaciji slike in opisa, podobno kot tudi pozneje v sami aplikaciji. Preden pa se lotimo razteznih vaj, je dobro, da vsaj na kratko spoznamo terminologijo mišic in opredelimo mišične skupine, katere bomo zajeli v aplikaciji.

2.2.1 Pregled mišične terminologije

V aplikaciji bomo vaje opredelili na naslednje mišične skupine oz. mišice:

- Vrat,
- roke,
- hrbet,
- prsa
- zadnjica,
- zadnja loža,
- kvadriceps,
- biceps in
- triceps.

Ker je terminologija za nekatere mišice bolj samoumevna kot za druge, sledi kratka obrazložitev.

Biceps - dvoglava nadlaktna mišica, ki skrbi za krčenje komolčnega sklepa

Triceps - troglava nadlaktna mišica, ki skrbi za iztegovanje komolčnega sklepa

Kvadriceps - štiriglava sprednja stegenska mišica, ki skrbi za iztegovanje kolena

Zadnja loža - zadnja stegenska mišica, ki skrbi za krčenje kolena in iztegovanje kolka

2.2.2 Raztezanje vratnih mišic

Pokončno sedemo na klop ali stol. Z eno roko zagrabimo spodnji rob našega sedišča. Glavno nagnemo rahlo naprej in v stran v smeri proste roke. S to roko si lahko pomagamo pri krepitvi raztezanja in sicer tako, da se primemo za glavo in rahlo potiskamo diagonalno navzdol, proč od roke s katero držimo spodnji rob sedišča. Vajo ponovimo še v drugo stran.



Slika 2.2: Raztezanje vratnih mišic

2.2.3 Razteg bicepsa v opori

Pokončno se postavimo ob steno ali drog in v višini ramen stegnjeno roko naslonimo na steno tako, da naslonimo z zgornjim delom palca in kazalca. Obrnemo se stran od stene do približno pravega kota, nato pa intenzivnost raztezanja prilagajamo z dodatnim obračanjem vstran. Vajo ponovimo še z drugo roko.

2.2.4 Razteg tricepsa spredaj

V vzravnani stoji eno roko iztegnemo pred sabo v višini ramen. Z drugo roko poprimemo za komolec in ga vlečemo k sebi. Ko opravimo s prvo roko, vajo ponovimo še z drugo roko.



Slika 2.3: Razteg bicepsa v opori



Slika 2.4: Razteg tricepsa spredaj



Slika 2.5: Razteg tricepsa ob steno

2.2.5 Razteg tricepsa ob steno

Stopimo do stene. Obrnjeni smo proti steni, ena noga naj bo rahlo nazaj, druga rahlo pokrčena. Roko, ki je bližje steni, dvignemo navpično navzgor in spustimo podlaket za glavo. S komolcem se naslonimo na steno in počasi raztezamo. Druga roka naj služi kot pomoč pri opori. Vajo nato ponovimo še z drugo roko.

2.2.6 Razteg prsne mišice s pokrčenim komolcem

Postavimo se blizu stene ali druge opore. Roko v komolcu pokrčimo ter dlan in komolec naslonimo na oporo. Komolec naj bo nekje v višini ramen, hrbet pa naj bo vzravnani. Obrnemo se v stran od stene. Za zviševanje intenzitete raztezanja se po občutku obračamo še bolj v stran od stene. Vajo nato ponovimo še z drugo roko.



Slika 2.6: Raztezanje prsne mišice s pokrčenim komolcem

2.2.7 Razteg prsne mišice z iztegnjenim komolcem

Zopet se postavimo blizu stene ali druge opore. Roko, ki je bližje steni, iztegnemo in dlan položimo na steno. Roka naj bo v višini ramen, hrbet pa naj bo vzravnani. Po občutku se obračamo v stran od stene. Vajo ponovimo za obe roki.

2.2.8 Razteg mečne mišice

Eno nogo potisnemo daleč nazaj in peto upremo v tla. Sprednja noga je rahlo pokrčena. Na stegno prednje noge ali kakšno drugo oporo se opremo z dlanmi. Zgornji del telesa rahlo nagnemo naprej, pri tem pa ohranimo vzravnani hrbet. Za krepitev raztezanja potiskamo boke naprej. Vajo ponovimo še z drugo nogo.



Slika 2.7: Razteg prsne mišice z iztegnjenim komolcem



Slika 2.8: Razteg mečne mišice



Slika 2.9: Raztezanje mečne mišice s pokrčeno nogo

2.2.9 Razteg mečne mišice s pokrčeno nogo

Nogi rahlo razkoračimo in obe pokrčimo, zadnjo bolj kot prednjo. Peta pri zadnji nogi naj ostane na tleh. Roke damo v oporo na prednje stegno ali na steno, mizo, ... Hrbet naj ostane raven. Za močnejše raztezanje poskušamo zadnjo nogo čim bolj pokrčiti in boke ter kolena potisniti naprej, toda peta naj ostane na tleh. Vajo ponovimo še z drugo nogo.

2.2.10 Dotik prstov stoje

Postavimo se z nogami v širini ramen ali ožje. V predklonu se nato poskušamo dotakniti tal ali pa se z rokami poskušamo prijeti za gležnje. Vaje ne izvajamo sunkovito, kolikor dosežemo, dosežemo. Pazimo na ravnotežje.



Slika 2.10: Dotik prstov stoje



Slika 2.11: Izpadni korak



Slika 2.12: Koleno na prsa

2.2.11 Izpadni korak

Na podlogi izvedemo izpadni korak tako, da položimo koleno na tla, drugo nogo pa v kolenu pokrčimo pred sabo in se upremo na stopalo. Težo počasi prenašamo naprej, hkrati pa naprej rinemo tudi boke. Vajo nato ponovimo še z drugo nogo.

2.2.12 Koleno na prsa

Uležemo se na udobno podlago. Ena noga naj rahlo pokrčena ostane na tleh, drugo nogo pa pokrčimo v kolenu in približamo k prsim. Z rokami primemo pokrčeno nogo pod kolenom in za večjo intenziteto vaje močnejše potegnemo k sebi. Vajo ponovimo še z drugo nogo.



Slika 2.13: Ležeči razteg zadnje lože

2.2.13 Ležeči razteg zadnje lože

Podobno kot pri prejšnji vaji, se uležemo na udobno podlago. Ena noga naj bo rahlo pokrčena, stopalo naj ostane na tleh. Drugo nogo poskušamo čim bolj stegniti in dvigniti. Z rokami primemo stegno in ga počasi vlečemo k sebi za večjo intenziteto vaje. Vajo ponovimo še z drugo nogo.

2.2.14 Sedeči enonožni razteg zadnje lože

Usedemo se na udobno podlago. Eno nogo iztegnemo pred sabo, drugo pa pokrčimo in podplat naslonimo na notranji del stegna iztegnjene noge. Z roko se poskušamo dotakniti prstov iztegnjene noge, hkrati pa pazimo, da ohranimo hrbet kar se da vzravnane. Vajo ponovimo še z drugo nogo.



Slika 2.14: Sedeči enonožni razteg zadnje lože



Slika 2.15: Sedeči sonožni razteg zadnje lože



Slika 2.16: Metuljček

2.2.15 Sedeči sonožni razteg zadnje lože

Podobno kot pri prejšnji vaji se usedemo na udobno podlago. Obe nogi iztegnemo pred sabo, z rokami pa se poskušamo dotakniti prstov na nogah. Za krepitev raztezanja poskušamo prste na nogah pritegniti k telesu. Hrbet naj ostane kar se da vzravn.

2.2.16 Metuljček

Sedemo na udobno podlago. Noge v kolenih pokrčimo in postavimo tako, da sta podplati naslonjena druga na drugo. Hrbet naj ostane vzravn. Z rokami se primemo za stopala in za krepitev raztezanja počasi vlečemo k sebi, s komolcema pa rahlo pritiskamo na notranji del stegen.



Slika 2.17: Mačji razteg

2.2.17 Mačji razteg

Za raztezanje spodnjega dela hrbta se na udobno podlago postavimo na vse štiri. Tokrat za spremembo hrbet kolikor moremo usločimo navzgor.

2.2.18 Pogled v nebo

Na udobno podlago pokleknemo na kolena. Stopala previdno podstavimo pod zadnjico, s katero se naslonimo na pete. Vzravnani se počasi nagibamo nazaj in usmerimo pogled v nebo.

2.2.19 Gleženj na koleno

Uležemo se na udobno podlago in pokrčimo noge. Gleženj ene noge postavimo na koleno druge noge, z rokami pa primemo spodnjo nogo pod kolenom in jo po občutku vlečemo k sebi.



Slika 2.18: Pogled v nebo



Slika 2.19: Gleženj na koleno



Slika 2.20: Koleno preko telesa

2.2.20 Koleno preko telesa

Udobno se uležemo na hrbet. Ena noga naj ostane stegnjena, drugo pa v kolenu pokrčimo in prevrnemo čez prvo nogo. Z eno roko si pomagamo potiskati prevrnjeno nogo proti tlem, druga roka naj na drugi strani ostane stegnjena na tleh. Hrbet in ramena poskušamo zadržati na tleh. Vajo ponovimo še z drugo nogo.

2.2.21 Ležeči razteg kvadricepsa

Udobno se uležemo na tla in se obrnemo na bok. Glavo položimo na roko, ki je iztegnjena pod glavo. Zgornjo nogo v kolenu pokrčimo nazaj, s prosto roko pa primemo gleženj in ga previdno pritegnemo k zadnjici. Stegna naj bosta čim bližje. Za okrepitev raztezanja bok počasi potiskajmo naprej. Pri tem pazimo, da ne usločimo ledvenega dela hrbtenice. Vajo ponovimo še z drugo nogo.



Slika 2.21: Ležeči razteg kvadricepsa

2.3 Kaj pričakujemo od aplikacije

Sedaj, ko smo spoznali, kako, kdaj in zakaj se izvaja raztezanje, imamo boljši občutek o tem, kako naj bi nam aplikacija pomagala pri izvajanju le tega. Tako smo željene funkcionalnost oziroma specifikacijo zahtev aplikacije oblikovali v uporabniške zgodbe, ki opisujejo posamezne situacije.

2.3.1 Uporabniška zgodba 1: Izbor vaj

Uporabnik lahko prosto izbere vaje, katere želi izvajati. Izbor vaj lahko omeji na posamezne mišične skupine.

2.3.2 Uporabniška zgodba 2: Zvočna opozorila

Uporabnika naj aplikacija z zvočnim opozorilom opozori, ali gre za začetek ali konec vaje. Zvočni opozorili se naj razlikujeta.

2.3.3 Uporabniška zgodba 3: Časovni intervali

Izvajanje posamezne vaje naj bo časovno omejeno, prav tako tudi vmesni odmor. Uporabnik lahko dolžino teh časovnih intervalov nastavlja poljubno.

2.3.4 Uporabniška zgodba 4: Priljubljene vaje

Uporabnik lahko posamezno vajo označi kot priljubljeno. Priljubljene vaje naj bodo hitro dostopne.

2.3.5 Uporabniška zgodba 5: Različne naprave

Aplikacija naj se optimalno izriše tako na mobilnikih kot tudi tabličnih računalnikih, ne glede na velikost in resolucijo naprave.

2.3.6 Uporabniška zgodba 6: Navodila za raztezanje

Uporabnik ima na voljo splošna vodila za raztezanje, lahko pa si prebere tudi tekstovni opis posamezne vaje.

Poglavje 3

Opis uporabljenih orodij in tehnologij

Do sedaj smo spoznali teorijo na kateri bo slonela naša aplikacija, zapisali pa smo tudi, katere funkcionalnosti moramo realizirati. Manjka nam le še opis orodij in tehnologij s pomočjo katerih bo naša aplikacija zaživel.

3.1 Android

Android¹ je odprtokodni mobilni operacijski sistem [12], zgrajen na osnovi Linuxovega jedra. Za njegov razvoj je najbolj zaslužen Google, ki je ustanovil poslovno združenje več podjetij, imenovano Open Handset Alliance (OHA) ter prevzel hitro rastoče podjetje Android Inc. Javnosti so združenje predstavili v začetku novembra 2007, med cilje pa so si zastavili prizadevanje k razvoju odprtokodnih standardov na področju telefonije ter ostalih prenosnih naprav ter stremenje k razvoju vedno boljših in cenejših naprav. S tem so želeli te naprave približati čim več uporabnikom, med njimi tudi tistim s tanjšo denarnico.

Snovalci so se sprva lotili razvoja naprednejšega operacijskega sistema za kamere. Ko so spoznali, da je ta trg za njih enostavno premajhen, so svoje

¹<http://www.android.com/>



Slika 3.1: Droid robot, logotip Androida

moči preusmerili v razvoj mobilnega operacijskega sistema, ki bi kljuboval takrat prevladajočima operacijskima sistemoma Symbian in Windows Phone.

S predstavitvijo 7. novembra 2007 in splavitvijo prvega telefona z operacijskim sistemom Android se je pričel pohod na vrh najbolj uporabljanega mobilnega operacijskega sistema. Štiri leta kasneje, 15. novembra 2011, je Android presegel že kar polovično prisotnost na vseh pametnih telefonih.

Android je vpeljal kar nekaj prednosti. Ker gre za odprtokoden sistem je razvijanje programov lažje in cenejše. Velik del programske opreme je tako dostopen brezplačno. Android prinaša cenejše in enostavnejše razvijanje pametnih telefonov, saj proizvajalcem ni več potrebno razvijati operacijskih sistemov. Tako se lahko osredotočijo le na razvoj posameznih komponent sistema. Android je za uporabo preprost in odziven, omogoča pa tudi večopravilnost. Googlovim uporabnikom je na roko pisano tudi samodejno sinhroniziranje z Googlovimi storitvami.

Da gre za izredno prilagodljiv operacijski sistem, priča širok spekter naprav, katere poganja Android. Android tako najdemo na najrazličnejših mobilnih telefonih, tabličnih računalnikih, razvoj pa gre tudi v smeri avtomobilske industrije, televizijskih sprejemnikov in nosljivih naprav, kot so pametne ure.

Izbiro operacijskega sistema Android lahko še potrdimo z rezultati² raziskave tržnega deleža, ki jo je opravilo podjetje International Data Corporation (IDC). IDC je mednarodno podjetje, ki se ukvarja z analitiko in zbiranjem podatkov. Po podatkih njihove raziskave je Android v letošnjem letu dosegel že preko 80 odstotni delež na trgu. Androidne naprave se tako še vedno uspešno širijo in beležijo vse večji delež.

3.2 Android SDK

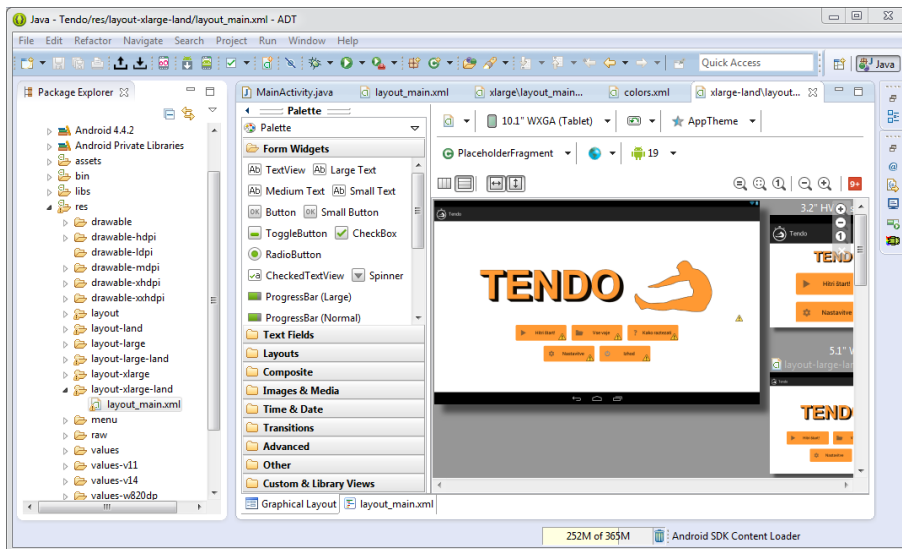
Android SDK [13] je paket programske opreme, ki vsebuje celovit komplet orodij za razvijanje aplikacij za platformo Android. Vključuje dokumentacijo, razhroščevalnik, knjižnice in primere programske kode. Na voljo je za vse večje operacijske sisteme. Tesno združljiv je z razvijalskim okoljem Eclipse. Paket se sproti posodablja vzporedno s splošnim razvojem platforme Android tako, da so razvijalcem vedno na voljo najnovejše zmožnosti, ki jih ponujajo naprave z Androidom.

3.3 Android developer tools

Za programiranje bo uporabljeno razvijalsko orodje Android developer tools (ADT)³. ADT je v bistvu razširitev in prilagoditev obstoječega razvijalskega orodja Eclipse. Gre za odprtokodno, zastojno in celovito orodje, katerega lahko poganjamo na vseh večjih operacijskih sistemih. Ponuja nam napredne zmožnosti za izdelavo, testiranje, razhroščevanje in pakiranje aplikacij. Ponuja tako hitre popravke kot delo z ukazno vrstico, razhroščevanje preko povezave usb in beleženje podatkov kar s telefona na računalnik. Odlikuje ga tudi delo z grafičnimi gradniki, s katerimi lahko brez težav z enostavno kombinacijo povleci-spusti gradimo še tako zapletene aplikacije. Omogoča nam testiranje na pravih napravah preko usb povezave, s pomočjo virtualnih

²<http://www.idc.com/getdoc.jsp?containerId=prUS25037214>

³<http://developer.android.com/tools/help/adt.html>



Slika 3.2: Zaslonski posnetek programa Eclipse, razširjen z ADT

naprav pa lahko simuliramo poljubne telefone ali tablice.

3.4 Programski jezik Java

Java je objektno orientiran programski jezik, ki ga je razvilo podjetje Sun Microsystems leta 1995 [14]. Poslučuje se načela WORA, »write once, run anywhere«, kar pomeni lahko prenosljivost med različnimi sistemi, torej programiraj enkrat in brez težav poženi na več različnih sistemih.

Pri razvoju tega programskega jezika so sledili petim osnovnim zahtevam citeaboutjava:

- Naj bo preprost in objektno orientiran,
- naj bo robusten in varen,
- naj bo preprosto prenosljiv,
- naj se izvaja učinkovito,
- in naj bo dinamičen in omogoča večnitno delovanje.

Javo delimo na tri različice. Prva, J2SE, je standardna različica, ki je namenjena osebnim računalnikom. Druga, J2ME, je namenjena mobilnim napravam. Poznamo pa še tretjo, J2EE(enterprise edition), ta pa je poslovna različica Jave.

3.5 Podatkovni sistem SQLite

SQLite⁴ je podatkovno relacijski sistem, lociran v majhni knjižnici oziroma datoteki manjše velikosti [15]. Gre za samostojno bazo, ki je integrirana v sam projekt v katerem jo uporabljamo. Glede na ostale baze, ta torej ni ločena od projekta in se ne nahaja na oddaljenem strežniku, do katerega bi bilo potrebno dostopati. Temu primerno je zato hitro in enostavno dosegljiva in zato vse pogostejša izbira pri razvijanju programske opreme.

Sintaksa je v grobem podobna SQL podatkovnim sistemom z nekaj manjšimi razlikami. Ker gre za enostavno realizacijo baze, omogoča samo en hkraten dostop naenkrat za pisanje, bralne operacije pa se lahko izvajajo tudi vzporedno.

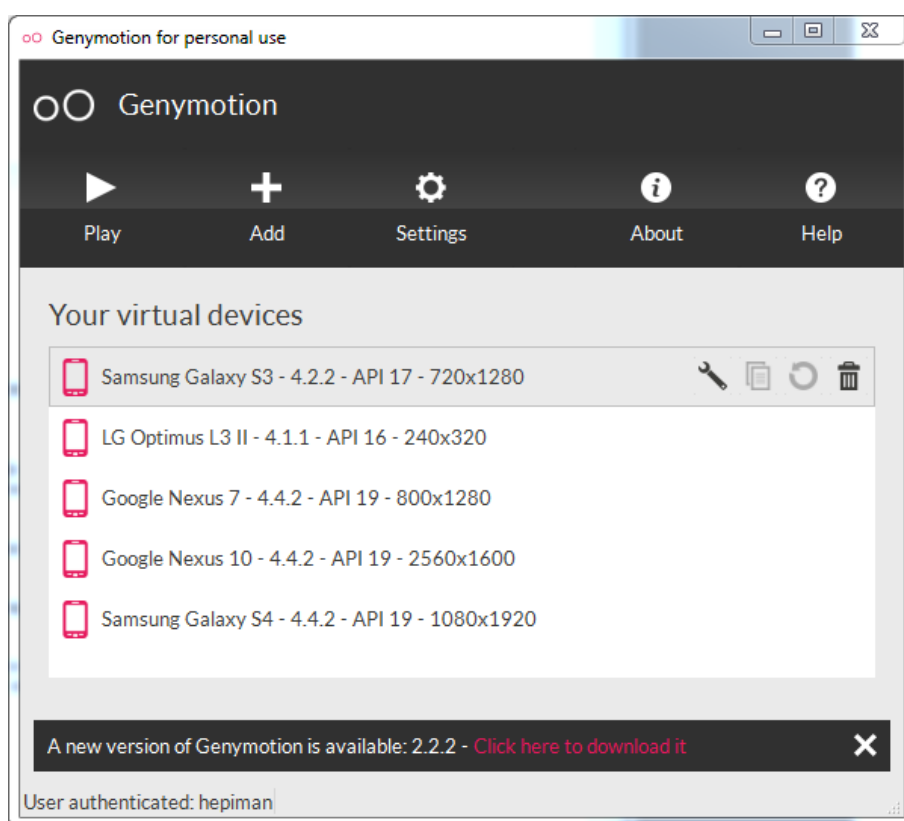
Sistem slovi po svoji zanesljivosti, saj je pred vsako novo izdajo temeljito pregledan, s čimer razvijalci preprečijo napake.

3.6 GenyMotion

Ker je simulacija naprav v orodju ADT izjemno počasna, bomo posegli po programu Genymotion⁵. Gre za hiter in enostaven simulator, katerega brez težav povežemo tudi z ADT. Simuliramo lahko več naprav naenkrat. Izbiramo lahko med širokim naborom naprav, vse od najmanjših telefonov do večjih tabličnih računalnikov, kar je idealno ko želimo preveriti kompatibilnost naše aplikacije na raznovrstnih napravah, pa vseh teh nimamo doma.

⁴<http://www.sqlite.org/>

⁵<http://www.genymotion.com/>



Slika 3.3: Zaslonski posnetek programa Genymotion

Poglavje 4

Programska rešitev mobilne aplikacije

4.1 Poimenovanje in logotip

Da bo aplikacija že na prvi pogled pritegnila uporabnika, potrebuje dober logotip in izvirno ime. Zavaljo izstopanja in preprostosti smo ime iskali kar v latinščini. Raztezanju se po latinsko reče *tendens*, to besedo pa smo okrajšali in dobili Tendo.

Glede na to, da bo naša aplikacija združevala funkcijo štoparice ter raztezanja smo obliko iskali s tem izhodiščem. Vzeli smo obris štoparice, vanjo pa smo postavili silhueto človeka, ki izvaja raztezanje.

Tako že s samim imenom in logotipom povzamemo vsebino in namen naše aplikacije. V poplavi mobilnih aplikacij, ki jih najdemo na tržnici aplikacij Google Play¹, je izstopajoč logotip pomemben ključ do uspeha.

4.2 Nastavitve ADT

Sprotni razvoj programskega paketa Android SDK prinaša številčenje posameznih verzij. Tako se funkcionalnosti neprestano širijo. Vsaka nova različica

¹<https://play.google.com/>



Slika 4.1: Logotip aplikacije Tendo

Androida ima tako zapisano številko, katero različico Android SDK uporablja [16]. Android 4.4 Kitkat² tako uporablja najnovejši paket s številčno oznako 19. Tako so tistim napravam, ki so zmožne poganjati novejši paket, na voljo novejša funkcionalnosti, ki jih naprave s starejšo različico Androida ne morejo koristiti. Ko se lotevamo izdelave mobilne aplikacije, moramo tako najprej nastaviti dve številčni vrednosti:

- Minimum SDK version - označuje najnižji paket opreme, ki nam še omogoča vse funkcionalnosti za normalno poganjanje naše aplikacije
- Target SDK version - označuje številčno oznako paketa, za katerega je bila aplikacija namenjena in testirana, sicer pa naj bi delovala vse do zgoraj omenjene najnižje dovoljene različice.

V našem primeru smo izbrali različico paketa označeno s številko 14, kar zajame Androidne naprave z operacijskim sistemom različice 4.0 in navzgor.

²<http://www.android.com/versions/kit-kat-4-4/>

Za to smo se odločili, saj je naprav, ki uporabljajo starejšo operacijskega sistema le še 14,3 odstotka³, ta delež pa vztrajno pada.

Za ciljno različico smo izbrali kar trenutno najnovejši dostopen paket, ta je označen s številko 19 in pripada različici Androida 4.4 z oznako KitKat.

4.3 Podatkovna baza

Preprosto podatkovno bazo za našo aplikacijo smo si zamislili s tremi glavnimi tabelami. Podatkovno bazo tako sestavljajo naslednje tabele:

- Exercise
- Tag
- ExerciseTag

Relacija tabel je prikazana s sliko 4.2. Shemo smo izrisali s pomočjo spletne storitve Online database schema designer⁴.

Kot je razvidno iz sheme podatkovne baze, lahko v tabelo Exercise zapišemo ime in opis vaje, lokacijo oziroma ime slike ter dve oznaki. Prva oznaka je namenjena označevanju dvostranskosti, ki nam na primer pove ali moramo vajo izvesti najprej z levo roko in nato še z desno. Z drugo oznako pa lahko uporabnik označi vajo označi ali gre za priljubljeno vajo ali ne.

Tabela Tag vsebuje le ime oznake oziroma kategorije, ki je pozneje uporabljeno pri razvrščanju razteznih vaj.

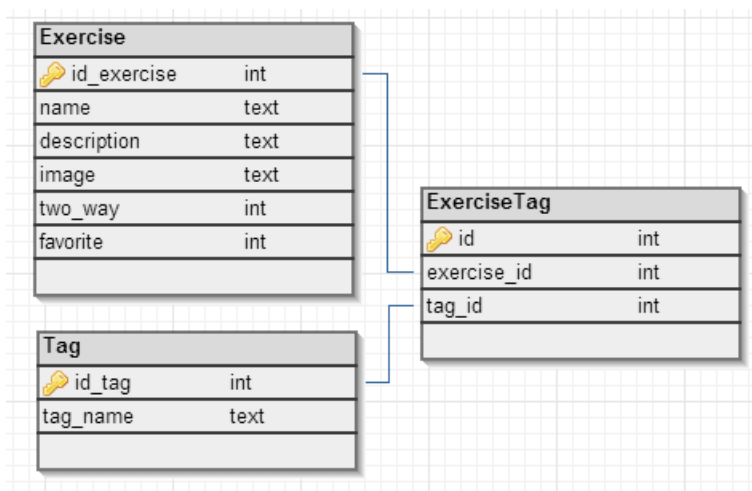
Tabela ExerciseTag je vmesna povezovalna tabela, ki povezuje vaje in kategorije.

4.4 Realizacija uporabniške zgodbe 1

Prva uporabniška zgodba, najdemo jo na strani 24, se navezuje na prost izbor vaj.

³<https://developer.android.com/about/dashboards/>

⁴<http://dbdsgnr.appspot.com/>



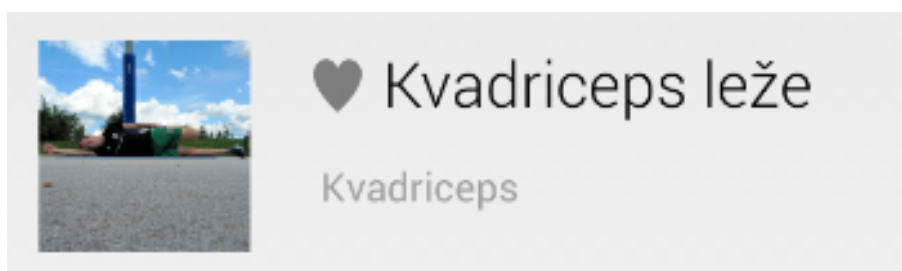
Slika 4.2: Shema podatkovne baze za aplikacijo Tendo

Naša aplikacija ponuja nabor 20 vaj za raztezanje. Uporabniku smo omogočili prost izbor tako, da smo vaje zvrstili s pomočjo gradnika ListView. Gre za gradnik, ki ponuja vrstični izpis elementov, pri čemer vsaka vrstica predstavlja ločen element.

Podatke za prikaz gradniku posredujemo s pomočjo Data Adapterja, ki ni nič drugega kot vmesni del, ki gole podatke hrani v obliki, katero je ListView zmožen prikazati. Ta adapter je tudi tisti, ki upravlja s podatki in poskrbi za njihov izpis.

Android ima sicer nekaj prednastavljenih osnovnih izgledov. Eden izmed teh je sicer sestavljen iz slike in priloženega teksta, to pa za nas še vseeno ni bilo dovolj. Zato smo razvili prilagojen adapter. Kot je razvidno iz slike 4.3, smo izgled posamezne vrstice oblikovali po meri. Tako na levi prikažemo pomanjšano sliko vaje, zraven pa izpišemo naslov vaje ter pod naslovom še kategorije, v katere spada posamezna vaja. Glede na to, ali je vaja označena kot priljubljena ali ne, se tu pred imenom vaje izriše še znak za priljubljeno vajo.

Znotraj tega adapterja pa poskrbimo tudi za označevanje vaj, katere sproti izberemo. To poskrbimo tako, da izbrani vaji spremenimo barvo



Slika 4.3: Izgled posamezne vrstice v gradniku ListView

ozadja. Za enostavnejšo uporabniško izkušnjo smo dodali tudi gumb za izbor vseh vaj naenkrat.

Da ob večji količini vaj enostavneje najdemo prave, lahko nabor vaj s pomočjo gradnika Spinner omejimo na posamezno mišično skupino ali pa prikažemo priljubljene vaje.

4.4.1 Optimizacija vrstičnega prikaza

Poleg izpisa podatkov smo tu poskrbeli še za optimizacijo le tega. Ukaz `findViewById()` namreč išče posamezne gradnike v izgledu vrstice, katere moramo nastaviti, ta ukaz pa je zelo zamuden. V kolikor vseh vrstic ne moremo naenkrat prikazati na eni strani, lahko z drsanjem navzdol prikažemo več vrstic. Tako se nam vrhnje vrstice skrijejo, pokažejo pa se nove. Problem nastane pri večji količini teh vrstic, saj ob kopici počasnih klicev postane drsanje počasno in zatikajoče. Z optimizacijo se lotimo ravno tega problema. V kolikor smo naše delčke v izgledu vrstice že locirali, to ustrezno shranimo in nato uporabimo ponovno. Pomagamo si tako, da ustvarimo nov razred, ki bo hranil podatke o gradnikih posamezne vrstice, kot je to razvidno s primera 4.1.

```
class MyViewHolder {  
    ImageView image;  
    TextView title;
```

```
TextView desc ;

MyViewHolder(View v) {
    image = (ImageView) v.findViewById(R.id
        .exerciseImageViewSingleRow) ;
    title = (TextView) v.findViewById(R.id .
        exerciseNameViewSingleRow) ;
    desc = (TextView) v.findViewById(R.id .
        exerciseTagsViewSingleRow) ;
}
}
```

Izsek programske kode 4.1: Razred za hrambo gradnikov

Ko pozneje nastavljam gradnike posamezne vrstice, najprej potrebujemo vrstico. V kolikor vrstico uporabljamo prvič, jo bomo zgradili od začetka in jo shranili v naš razred za hranjenje izgleda vrstice. V kolikor vrstico ne uporabljamo prvič, jo potegnemo iz našega razreda ter le še nastavimo ustrezne podatke, saj so gradniki vrstice že locirani hitro dostopni. S tem prihranimo precej časa, ki bi ga sicer zapravili z vsakokratnim iskanjem gradnikov posamezne vrstice.

```
if (row == null) {
    LayoutInflater inflater = (LayoutInflater)
        context
            .getSystemService(Context .
                LAYOUT_INFLATER_SERVICE) ;
    row = inflater.inflate(R.layout.single_row ,
        viewGroup , false) ;

    holder = new MyViewHolder(row) ;
    row.setTag(holder) ;

} else {
```

```
        holder = (MyViewHolder) row.getTag();  
    }
```

Izsek programske kode 4.2: Uporaba hrambnega razreda

4.5 Realizacija uporabniške zgodbe 2

Druga uporabniška zgodba, najdemo jo na strani 24, se navezuje na zvočna opozorila.

Za začetek vaje smo izbrali kratek zvok enega udarca, za konec pa zvok trojnega udarca po zvoncu. Obe zvočni datoteki sta oblike .wav, predvajamo pa jih s pomočjo objekta `MediaPlayer`, ki je del programskega paketa `Android`. Za predvajanje zvoka tako le inicializiramo instanco objekta `MediaPlayer`, ji podamo primeren kontekst in pot do zvočne datoteke, nato pa poženemo z ukazom `start`. Tako enojni kot trojni zvonček smo zapakirali v metodo, ki jo tako pokličemo ob ustreznem času, torej začetku ali koncu vadbe.

```
public void startBell() {  
    MediaPlayer mPlayer = MediaPlayer.create(  
        getBaseContext(), R.raw.bell1);  
    mPlayer.start();  
}  
  
public void stopBell() {  
    MediaPlayer mPlayer = MediaPlayer.create(  
        getBaseContext(), R.raw.bell3);  
    mPlayer.start();  
}
```

Izsek programske kode 4.3: Metodi za zvočno opozorilo

Zvočna opozorila smo kombinirali s tekstovnimi. Glede na to, da uporabnik spremlja kaj se dogaja na zaslonu, bo med drugimi podatki zaznal tudi tekst, ki ga opozarja, naj se pripravi na naslednjo vajo.

4.6 Realizacija uporabniške zgodbe 3

Tretja uporabniška zgodba, katero najdemo na strani 25, opisuje izbiro dolžine časovnega intervala za izvajanje vaje ter dolžino vmesnega odmora.

Najprej smo morali rešiti problem merjenja pretečenega časa. Pomagali smo si z metodo `System.currentTimeMillis`, ki vrača čas v milisekundah. Ob začetku vaje smo zabeležili trenutni čas, sproti smo v intervalu vsake desetinke sekunde beležili pretečen čas, hkrati pa posodabljali števec časa na zaslonu. Ko je pretečen čas dosegel prednastavljen čas vaje, smo sprožili zvočno opozorilo.

Beleženje časa in posodabljanje podatkov na zaslonu smo opravljali s pomočjo zakasnenih akcij, ki smo jih sprožali s pomočjo metode `postDelayed`. S to metodo lahko določimo kateri del programske kode se bo izvedel z določenim časovnim zamikom od tega trenutka, kot je to razvidno na primeru 4.4. Ta del kode prikazuje, kako po treh sekundah (3000 ms) poženemo časovni števec ter sprožimo zvočno opozorilo za začetek vadbe.

```
startExerciseHandler.postDelayed(new Runnable() {  
    @Override  
    public void run() {  
        startTimer(findViewById(android.R.id.  
            content).getRootView());  
        singleBellRing.run();  
    }  
}, 3000);
```

Izsek programske kode 4.4: Primer uporabe metode `postDelayed`

V ozadju se torej skriva kompleksna koda, ki ob vsakem preverjanju izpolnjenih pogojev z zamikom sproži naslednje dejanje, pa naj bo to nova vaja, druga stran vaje, ali pa le osveži tekst, ki prikazuje preostanek časa.

Sedaj, ko smo uredili merjenje časa, moramo le še uporabniku omogočiti nadzor nad trajanjem izvedbe vaje in vmesnega odmora. V aplikacijo smo dodali pogled Nastavitve, kjer lahko uporabnik poljubno izbere časovne inter-

vale. Nastavitve beležimo s pomočjo **SharedPreferences**, ki kaže na zbirko podatkov, organizirano v parih ključ - vrednost.

Kot je razvidno iz primera 4.5, s pomočjo upravljalca dobimo preference, nato pa posamezne nastavitve dobimo z iskanjem po ključu. Za pretvorbo sekund v milisekunde vrednosti pomnožimo še s 1000, nato pa so pripravljene za uporabo. S 1000 moramo pomnožiti zato, ker moramo metodi **postDelayed** čas podati v milisekundah.

```
SharedPreferences preferences = PreferenceManager
    .getDefaultSharedPreferences(getApplicationContext());
duration = preferences.getInt("exerciseTime", 10) *
    1000;
pauseTime = preferences.getInt("pauseTime", 5) * 1000;
```

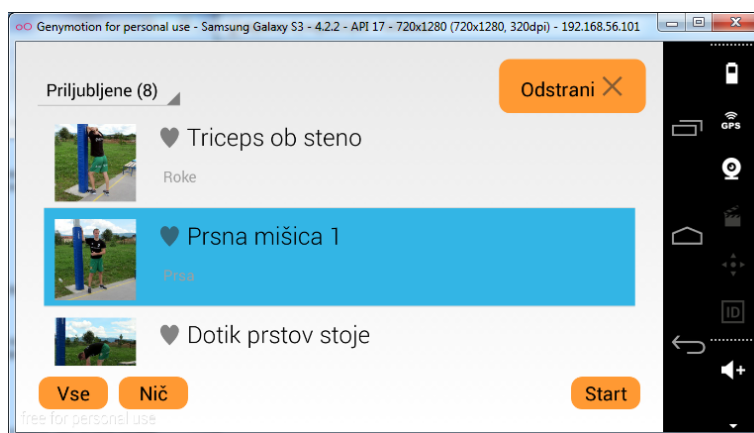
Izsek programske kode 4.5: Branje shranjenih nastavitev

V kolikor uporabnik posebej ne nastavlja časovnih intervalov, je čas trajanja vaje privzeto nastavljen na 10, čas odmora pa na 5 sekund.

4.7 Realizacija uporabniške zgodbe 4

Urejanje in hiter dostop priljubljenih vaj je bil enostaven za realizacijo. Že v samem začetku smo vajam v podatkovni bazi določili poseben indikator, ki beleži ali je vaja označena kot priljubljena ali ne. Uporabnik tako v pogledu, kjer ima na voljo nabor vseh vaj, izbere nekaj vaj in jih nato z gumbom označi kot priljubljene. V pregledu priljubljenih vaj (slika 4.4) jih na podoben način lahko tudi odstrani.

Hiter dostop smo uredili z gumbom na začetnem zaslonu aplikacije z imenom **Hitri štart**. Pritisk na ta gumb požene vadbo z vsemi vajami, ki so označene kot priljubljene.



Slika 4.4: Zaslonski posnetek priljubljenih vaj

4.8 Realizacija uporabniške zgodbe 5

Peta uporabniška zgodba, najdemo jo na strani 25, govori o pestrem naboru naprav z različnimi velikostmi zaslonov.

Za optimalen prikaz na napravah različnih velikosti moramo paziti na več stvari [17].

Velikosti zaslonov v grobem delimo na štiri kategorije, na **majhne**, **normalne**, **velike** in **zelo velike** zaslone. Podobno jih delimo tudi glede na količino pikslov, ki so jih naprave sposobne prikazati. Tako poznamo naprave ki imajo **majhno**, **srednjo**, **visoko** in **zelo visoko** gostoto prikaza.

Android sicer avtomatično razteza elemente na večjih zaslonih, toda za res optimalen prikaz moramo za vsako kategorijo velikosti pripraviti svoj pogled, oblikovan in shranjen v XML datoteki. Te datoteke moramo združevati v mape, katerim dodamo pripono s kategorijo, za katero so pripravljeni. Sistem nato glede na velikost zaslona vzame pogled iz prave mape. V kolikor pripravimo tudi ločene poglede glede na postavitev naprave, moramo prav tako pripraviti pogled in ga postaviti v mapo s pravo pripono. Če je pogled pripravljen za pokončno postavitev, pogled vstavimo v mapo brez pripone, če pa je pripravljen za vodoravno postavitev naprave, pa pogled vstavimo v mapo s pripono `-land`. Ne glede na to, za katero velikost zaslona pripra-

vljamo pogled, pa se morajo imena pogledov v vseh mapah ujemati.

```
Tendo/  
    res/  
        layout/  
            layout_main.xml  
        layout-land/  
            layout_main.xml  
        layout-large/  
            layout_main.xml  
        layout-large-land/  
            layout_main.xml  
        layout-xlarge/  
            layout_main.xml  
        layout-xlarge-land/  
            layout_main.xml
```

Izsek programske kode 4.6: Drevesna struktura pogledov

Na podoben način se različnim velikostim zaslonov prilagaja tudi kakovost slik. Za naprave, ki so zmožne prikazovati večjo gostoto pikslov, pripravimo kvalitetnejše slike, za slabše naprave pa pripravimo manjše slike in s tem pohitrino nalaganje, obenem pa ne nalagamo večjih slik po nepotrebnem. Slike morajo ohranjati enako poimenovanje. Naprave podobno kot v prejšnjem primeru z različnimi pogledi, tudi tokrat glede na zmožnost prikaza izberejo optimalno velike slike. V našem primeru prikazujemo malo slik, za te pa poskrbi sistem kar sam, saj velikost slik prilagodi avtomatično. Za sličice ob izbiranju vaj smo uporabili ustrezno pomanjšane različice slik.

4.8.1 Nabor naprav za testiranje

Za testiranje ustreznega prikaza smo aplikacijo testirali na naslednjih napravah.

Htc Desire X	
Resolucija	480x800
Gostota pikslov	233

Samsung Galaxy Tab 3 8.0	
Resolucija	800x1280
Gostota pikslov	189

Samsung Galaxy S3, Genymotion	
Resolucija	720x1280
Gostota pikslov	306

Samsung Galaxy S4, Genymotion	
Resolucija	1080x1920
Gostota pikslov	441

Asus Google Nexus 7, 2013, Genymotion	
Resolucija	800x1280
Gostota pikslov	216

Samsung Google Nexus 10, Genymotion	
Resolucija	2560x1600
Gostota pikslov	299

Večjih težav pri prilagajanju nismo imeli, saj smo med drugim uporabljali take gradnike, ki so neodvisni od velikosti zaslonov. Izdelovanje pogledov temelji na gnezdenju gradnikov. Korenski element določa način postavljanja gradnikov znotraj le tega. Širino in višino gradnikov lahko sicer določimo z natančnimi merami, lahko pa si pomagamo s konstantami kot sta `match_parent` in `wrap_content`. Prva poskrbi, da je gradnik enake velikosti kot njegov starš, druga pa velikost gradnika omeji glede na njegovo vsebino. Tako smo pri vseh večjih gradnikih uporabili `match_parent`, ki zavzame celotno površino starša. V primeru, da je gradnik korenski element, potem zavzame celotno površino zaslona in se s tem prilagodi velikosti zaslona.

Za realizacijo te zgodbe smo pripravili nekaj različnih pogledov za začetni pogled aplikacije. Glede na velikost smo prilagodili gumbe ter, glede na položaj v katerem je naprava, uporabili različen začetni logotip. Podobno se ob izvajanju vadbe glede na položaj naprave gradniki postavijo drugače.

4.9 Realizacija uporabniške zgodbe 6

Šesta uporabniška zgodba, ki jo najdemo na strani 25, govori pomoči uporabniku s tekstovnimi navodili za izvajanje razteznih vaj.

Navodila smo razdelili v dva dela in sicer na splošna navodila in navodila za posamezno vajo. Splošna navodila za raztezanje smo zbrali v ločenem pogledu, kateri je dostopen s prve strani aplikacije.

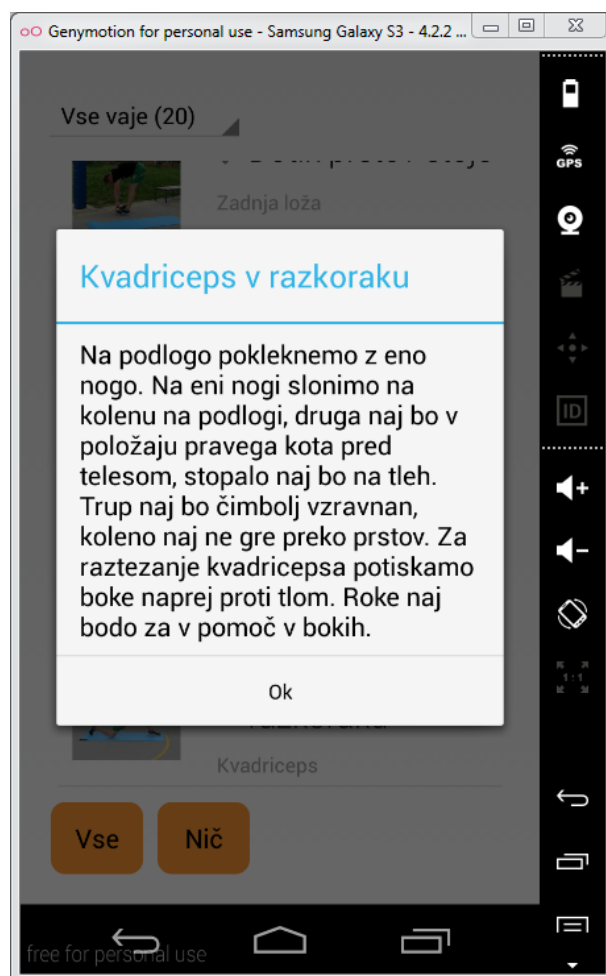
Sicer slike povedo veliko, pa vseeno smo uporabniku omogočili, da si prebere natančna navodila za izvajanje posamezne raztezne vaje. Ta podrobna navodila so dostopna z dolgim pritiskom na posamezno vajo, v pogledu, kjer so razvrščene vse vaje. Ob dolgem pritisku se pojavi okno z imenom vaje in tekstovnim opisom. Primer tega lahko vidimo na sliki 4.5.

4.10 Opis uporabe celotne aplikacije

Sled opis enkratne uporabe celotne aplikacije. Uporabnik je aplikacijo pognal prvič.

Ob prvem zagonu aplikacije uporabnika pričaka izskočno okno, v katerem se avtor uporabniku zahvaljuje za uporabo aplikacije ter mu svetuje, naj si najprej prebere splošna vodila za izvajanje raztezanja. S klikom na gumb OK se okno skrije in uporabnika pričaka začetni meni.

Najprej si ogleda splošna vodila za raztezanje, zatem pa nastavitve, katere pa za prvič pusti privzete. Vrne se v začetni meni, kjer je najbolj mamljiv gumb na katerem piše **Hitri štart!**. Ob kliku na ta gumb, izskoči obvestilo, ki pravi, da ni še nobena vaja označena kot priljubljena. Nadaljuje s pritiskom na gumb **Vse vaje**, ki odpre nov pogled z naborom vaj. Opazi, da so trenutno



Slika 4.5: Navodila za posamezno vajo

prikazane kar vse vaje. Na izbiro ima tudi kategorijo priljubljene, ta pa je zaenkrat prazna. Ob pregledu vaj spozna, da še ne pozna vseh vaj, zato na nekaterih vajah uporabi daljši pritisk, s čimer prikličemo izskočno okno s podrobnim opisom izvedbe vaje. Odloči se, da bo preizkusil kar vse vaje, zato uporabi priročen gumb s katerim označi vse vaje in vadbo požene s pritiskom na gumb **Start**. Izbor vaj se prenese v naslednje okno in prične se vadba.

Na zaslonu se prikaže ime vaje, slika vaje, čas nastavljen na 10 sekund in tekst, ki uporabnika opozarja naj se pripravi na vajo. Po treh sekundah se sproži zvočno opozorilo in čas se prične odšteti. Vadba nato poteka po naslednjem vrstnem redu, ki ga prikazuje Algoritem 1.

```

while št. trenutne vaje < št. vseh vaj do
    if pretečen čas > čas trajanja vadbe then
        sproži zvočno opozorilo za konec;
        sproži odmor;
        pripravi zvočno opozorilo za začetek;
        ponastavi števec za pretečen čas;
        if dvostranska vaja then
            | pripravi drugo stran, poženi nov timer;
        else
            | pripravi novo vajo, poženi nov timer;
        end
    else
        | posodablja čas;
    end
end

```

dialog - ponovi vadbo ali vrni v začetni meni aplikacije;

Algoritem 1: Potek vadbe

Zadovoljen z nekaterimi vajami, uporabnik zopet odpre spisek vaj, označi po želji in jih shrani kot priljubljene. V nastavitvah nato še poveča čas izvajanja ene vaje na 15 sekund, shrani in zapre aplikacijo.

Ob naslednjem zagonu tako preprosto uporabi hitri štart. Požene se

vadba, sestavljena iz priljubljenih vaj s 15 sekundnim trajanjem posamezne vaje.

Poglavje 5

Sklepne ugotovitve

V sklopu diplomskega dela smo izdelali mobilno aplikacijo za operacijski sistem Android, s katero nam mobilni telefon služi kot pripomoček pri raztezni vadbi.

Osredotočili smo se na pomemben, a pogosto zapostavljen del športne vadbe in sicer na raztezanje. Opisali smo kako raztezanje poteka in kako pozitivno učinkuje na človeka. Za aplikacijo smo zbrali nabor dvajsetih vaj, katere smo predstavili s slikami in tekstovnim opisom. Seznanili smo se z Androidom in potekom izdelovanja programske opreme za ta operacijski sistem. Vse skupaj smo povezali v uporabno aplikacijo. Ta uporabniku olajša izvajanje raztezne vadbe, saj mu ponuja izbor raztezni vaj, nato pa ga s pomočjo zvočnih opozoril in grafičnega prikaza vodi skozi samo vadbo.

Čeprav smo s samo izvedbo zadovoljni, vidimo v aplikaciji še mnogo možnosti za izboljšave.

- Širitev nabora vaj,
- zvočna opozorila, prilagojena posameznim vajam,
- posebni vadbeni programi npr. za osebe z manjšimi bolečinami v križu,
- dnevni opomniki za izvajanje raztezanja, ...

To so le nekatere izmed mnogih idej, katere bi v nadaljevanju močno nadgradile obstoječo aplikacijo. Konec koncev pa lahko izdelana aplikacija služi tudi

kot odlična osnova za pripravo podobne aplikacije, ki bo prav tako namenjena intervalnemu treningu.

Mobilni telefon tako lahko postane zelo dober športni pripomoček, saj je izjemno uporaben že v preprosti vlogi štoparice. Ko pa dodamo šopico senzorjev, pa postane dovršen pripomoček za športno udejstvovanje.

Z izdelavo te diplomske naloge in mobilne aplikacije pa sem izpolnil tudi osebni cilj. V sklopu študija se namreč nismo posebej ukvarjali z razvojem za Android, zato sem ob tej priložnosti izbral ravno to temo, da se seznanim še s tem. Android bo v prihodnosti gotovo ostal eden izmed vodilnih operacijskih sistemov, zato bo znanje s tega področja še kako dobrodošlo.

Literatura

- [1] B. Anderson, “Raztezanje za računalnikom in pisalno mizo”, 1998, str. 4–7.
- [2] C. A. Rahman, M. Schwarz, “Moški v formi”, 2007, str. 102–103.
- [3] P. Regelin, “Raztezanje”, 2005, str. 4–9.
- [4] Motorola DynaTAC. Dostopno na:
http://en.wikipedia.org/wiki/Motorola_DynaTAC
- [5] History of mobile phones. Dostopno na:
http://en.wikipedia.org/wiki/History_of_mobile_phones
- [6] Nokia 3310. Dostopno na:
http://en.wikipedia.org/wiki/Nokia_3310
- [7] Cell-ebration! 40 Years of Cellphone history. Dostopno na:
<http://mashable.com/2013/04/03/anniversary-of-cellphone/>
- [8] 10 Sensors of Galaxy S5: Heart Rate, Finger Scanner and more. Dostopno na:
<http://global.samsungtomorrow.com/?p=36031>
- [9] List of countries by number of mobile phones in use. Dostopno na:
http://en.wikipedia.org/wiki/List_of_countries_by_number_of_mobile_phones_in_use
- [10] Florida Survey Highlights Smartphone Statistics and Habits. Dostopno na:

<http://www.verizonwireless.com/news/article/2014/07/florida-survey-highlights-smartphone-use.html>

- [11] Mobile operating system. Dostopno na:
http://en.wikipedia.org/wiki/Mobile_operating_system
- [12] Android (operating system). Dostopno na:
[http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [13] Android software development. Dostopno na:
http://en.wikipedia.org/wiki/Android_software_development
- [14] Java (programming language) Dostopno na:
[http://en.wikipedia.org/wiki/Java_\(programming_language\)](http://en.wikipedia.org/wiki/Java_(programming_language))
- [15] About SQLite. Dostopno na:
<http://www.sqlite.org/about.html>
- [16] Android <uses-sdk>. Dostopno na:
<http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>
- [17] Supporting Multiple Screens. Dostopno na:
http://developer.android.com/guide/practices/screens_support.html
- [18] Raztezanje - stretching. Dostopno na:
<http://www.zdravstvena.info/preventiva/raztezanje-kdaj-se-raztezati-raztezovanje-kako-se-raztezamo-pravilno-raztezanje.html>